

## TP 1 PLSQL

Soit le schéma de base de données suivant :

Client : (**id**, prenom, nom, email, ville, );

Commande: (**id**, client\_id, date\_achat, reference);

commande\_ligne: (**id**, commande\_id, nom\_produit, quantite, prix\_total);

Produit : (**nom produit**, prix\_unitaire)

Les scripts de création des tables sont dans le repertoire

@\\iluvatar\export\documents\is3+4\sgbd\lekpa\1\_TD\_TP\_S3.sql

- 1- Mettre à jour la colonne Prix\_total de la relation commande\_ligne en implémentant la requête vue en TD.
- 2- Ecrire la procédure stockée sp\_InsertCommandeLigne(p\_id, p\_commande\_id, p\_nom\_produit, p\_quantite) qui permet d'insérer une nouvelle ligne dans la relation commande\_ligne.  
 La colonne Prix\_total doit être calculée avant insertion.  
 Pour le test, utiliser un id>120 pour ne pas violer la contrainte d'unicité.  
 Quels sont les cas d'erreurs ?
- 3- Modifier la procédure sp\_InsertCommandeLigne pour gérer les cas erreurs ;
- 4- Gestion des stocks : Ecrire la fonction fn\_QteProduitVendu(p\_nom\_produit) qui prend en paramètre le nom d'un produit et retourne la quantité de produit acheté. Faites un select sur commande\_ligne et retourner tous les produits et les quantités achetées. Afficher le resultat par ordre decroissant de vente.
- 5- Exécuter la fonction fn\_QteProduitVendu avec un nom de produit qui n'existe pas. Que remarquez-vous ?
- 6- Ecrire une nouvelle fonction fn\_PrixUnitaire(p\_nom\_produit) qui renvoie le prix unitaire d'un produit. Exécuter la fonction avec un nom de produit qui n'existe pas de deux façons différentes avec un nom de produit qui n'existe pas.
  - Dans un bloc pl/sql
  - Dans une requête sql en faisant « select fn\_PrixUnitaire (p\_produit) from dual »

Que remarquez-vous ? que peut-on conclure ?

- 7- Créer une fonction fn\_AfficheMontantFacture qui prend en paramètre l'id d'une commande et retourne le montant total de la facture.
- 8- Ecrire une fonction fn\_FraisLivraison(id\_commande) qui prend en paramètre l'id d'une commande et retourne le montant des frais de livraison. Utiliser la fonction précédemment créée et la structure de contrôle case when... then
  - 0 si commande >= 100 euros
  - 4 si commande >= 50 euros et < 100euros
  - 10 si commande < 50 euros.

Faites un select sur la table commande et afficher toutes les commandes et les frais de livraison.

- 9- Points de fidélité : Ecrire la procédure sp\_PointsFidélité(Id\_client) qui calcule les points de fidélité d'un client. 1 point de fidélité tous les 10 euros d'achat. Utiliser le curseur pour afficher tous les clients et leurs points de fidélité sous la forme : le client X a Y points de fidélité.